Mobile IPv6: Configuration and Trials

Tudor Blaga
Technical University of Cluj-Napoca
Tudor.Blaga@com.utcluj.ro

Virgil Dobrota
Technical University of Cluj-Napoca
Virgil.Dobrota@com.utcluj.ro

Daniel Zinca
Technical University of Cluj-Napoca
Daniel.Zinca @com.utcluj.ro

Mihai Vancea
Technical University of Cluj-Napoca
Mihai.Vancea@com.utcluj.ro

Abstract

The paper is focused on Mobile IPv6 recommendations, including draft-ietf-mobileip-ipv6-19 and mainly regarding to Linux Red Hat 8.0. Although Microsoft's Windows 2000/XP/2003 currently uses the obsolete recommendation draft-ietf-mobileip-ipv6-13, we did select also these operating systems for our trial. The testing scenario did not include from the beginning any wireless equipment. The minimal mobility demonstrator was based on three workstations in a wired network.

Note that as soon as the mobile node's connection is changing, the Mobile IPv6 specific procedures are starting: care-of-address auto-configuration, home registration, and CN notification. We were focused on preliminary evaluation of the following parameters: RTT (Round Trip Time), inter-arrival jitter and cumulative number of packets lost.

1. Introduction

Nowadays mobility support for Internet devices becomes more important, since mobile devices are getting more widespread. Furthermore cellular devices of the 3rd generation will be packet switched devices instead of circuit switched, therefore the need for Mobile IP increases.

Several problems arise, that make roaming with mobile Internet devices difficult. When roaming from one location to another, communication is not possible until the system configures a new IP address, the correct netmask and a new default router. The problem is caused by the routing mechanisms, which are used by IP. IP addresses define a kind of topological relation between the linked computers. The node's IP address identifies the link on which the node resides, as well as the node itself. If a node moves without changing its IP address, existing routing protocols are not able to deliver the datagrams to the new location.

Mobility Support in IPv6, called Mobile IPv6 [1] is designed to allow an IPv6 host to leave its home network without changing its address while maintaining all of its present connections and remaining reachable to the rest of the Internet. The mechanism is completely transparent to transport and higher-layer protocols and applications. The aim of this article is to present the main features of Mobile IPv6 and the configuration and testing of the Mobile IPv6 Linux and Windows implementations.

2. Mobility Support in IPv6

When a Mobile Node (MN) changes its point of attachment from one network to another it needs to change its IP address to a topologically correct one, to allow routers to divert datagrams to the new network address. However, at the same time other hosts communicating with MNs, called Correspondent Nodes (CN), need to be able to send packets to the MNs. The aim of Mobile IP is to solve this problem in a way that scales to large numbers of fast moving MNs.

Mobile IPv6 solves the routing problem caused by mobile users. It uses Home Agent (HA), which keeps track of the current care-of address (CoA), of the MN. CoA is the topologically correct address of the MN in the visited network. With this address HA can deliver datagrams that originally were sent to the MN's home address, by tunneling them to the CoA. When MN moves to another network, it informs the HA of its new CoA by sending a Binding Update (BU) message to the HA. The BU binds the new CoA to the home address of MN for a period of time. The information obtained from BUs is stored by the HA in a special data structure called binding cache.

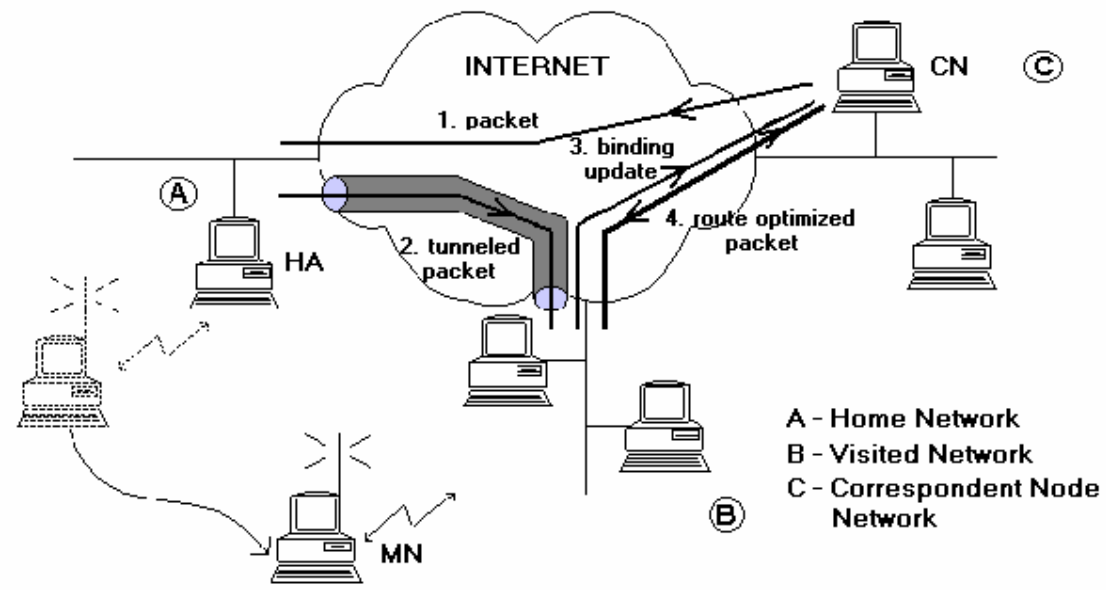

Figure 1. Mobile IPv6 Operation

The Mobile IPv6 operations are the following:

Step1: When the MN communicates with a CN, the CN uses the home address of the MN and the CN's packets are routed to the home network.

Step2: The packets are then tunneled to the MN by the HA, while the MN sends datagrams directly to the CN.

Step3: Because the home network is topologically far from the current location (visited network) of the MN, this is inefficient and this increases the load on the home network. The phenomenon is called *Triangle Routing* and Mobile IPv6 provides the solution to this, called route optimization. When a MN receives a packet tunneled by its HA it can determine that the original sender of the packet is not aware of the mobility of MN. To inform the CN, MN sends a BU to it.

Step4: This allows the CN to send datagrams directly to the MN's CoA, using a routing header. This procedure is illustrated by Figure 1.

If the delays between MN and its HA and CNs are large, hand-offs may lead to significant packet loss, especially on higher bandwidth links, such as WLANs. To perform smoother hand-offs a MN can also send a BU to its previous router, which will then act as a temporary HA and tunnel datagrams originally sent to the previous CoA to the new CoA of MN.

The signaling in Mobile IPv6 uses destination option headers, which are one type of IPv6 extension headers. They allow piggybacking of signaling information in packets carrying application data. There are four new types of destination options:

- Home Address option is used to carry the home address of MN, when it is away from its home network. It is necessary for allowing CN to demultiplex the datagrams it receives.
- Binding Update (BU) option creates, updates and deletes entries in the binding caches of HA and CN. It is used for creating a binding between the source address of the datagram and the home address in the home address option.
- Binding Acknowledgment (BA) option is sent by HA and by CNs in response to a BU to inform MN of the status of the binding update.
- Binding Request (BR) option is sent by CN to request a MN to refresh the binding cache entry for it.

3. Mobility and IPsec

BU and BA change state in the receiving nodes and thus they need to be authenticated. Especially BUs need to be authenticated as they remotely redirect the routing of datagrams to the home address of MN. Mobile IPv6 uses AH (IPsec Authentication Header), for this purpose. IPSec is a protocol designed to secure the TCP/IP protocol suite and it should be a part of every IPv6 implementation. AH is an IPv6 extension header, which protects the integrity of the whole datagram. So it also verifies the identity of the datagram's sender.

Although IPsec provides a means for authenticating the signaling it does not solve the problem of authorization. How can MN prove to a CN that it has the authority to change the routing of its datagrams? This is not a problem between MN and HA as they are likely to already have a Security Association. MN and CN may not have any knowledge of each other at the beginning of their communication and thus the setup of a SA is not trivial. The use of IKE (Internet Key Exchange) together with DNSsec provides a solution, with the assumption that both MN and CN use the same Public Key Infrastructure.

4. Mobile IPv6 for Linux

The package used for testing is MIPL (Mobile IPv6 for Linux). This is an implementation of Mobility Support in IPv6 developed by Helsinki University of Technology. The latest version, `mipv6-0.9.5-v2.4.20` implements `draft-ietf-mobileip-ipv6-19.txt`, although the current draft version is 21, and it works with `linux-2.4.20 kernel` release.

The package consists of a kernel module, a kernel patch and userspace programs for configuration and installation of the kernel part.

4.1. Installation

The MIPL kernel patch for the 2.4.20 version should only be applied against fresh kernel tree (or later, provided no changes have been made in the `net/ipv6` directory). It is called `mipv6-a.b-v2.4.x.patch` (where `a`, respectively `b` are the major and minor version numbers of MIPL and `x` is the kernel version sub-level against which the patch was made). MIPL has only been tested on RedHat 8.0 system but should work on any Linux system assuming that you already have a working 2.4.20 kernel and an IPv6 environment.

The first step is to install the kernel patch. Assuming you have a fresh 2.4.20 kernel tree in `/usr/src/linux` do the following:

```
cd /usr/src/linux
patch -p1 --dry-run < mipv6-a.b-v2.4.x.patch
```

This does not actually do anything but it displays errors if any. In case of errors, you should cancel the installation, otherwise you can type:

```
patch -p1 < mipv6-a.b-v2.4.x.patch
```

Now the kernel tree is ready for configuration. Run your favorite `make *config`. Make sure you have at least the following options set:

```
CONFIG_EXPERIMENTAL=y
CONFIG_SYSCTL=y
CONFIG_PROC_FS=y
CONFIG_MODULES=y
CONFIG_NET=y
CONFIG_NETFILTER=y
CONFIG_UNIX=y
CONFIG_INET=y
CONFIG_IPV6=m
CONFIG_IPV6_SUBTREES=y
CONFIG_IPV6_IPV6_TUNNEL=m
CONFIG_IPV6_MOBILITY=m
```

In the package you will find a script, `chkconf_kernel.sh` that can be used to check if you have configured the right options.

You may choose '`y`' instead of '`m`' if you don't want to build Mobile IPv6 as a module. The last configuration option is the newly added Mobility Support. By selecting this it enables Mobile IPv6 Correspondent node operation. You may also select following options:

```
CONFIG_IPV6_MOBILITY_MN
CONFIG_IPV6_MOBILITY_HA
CONFIG_IPV6_MOBILITY_DEBUG
```

The first two control whether you want to have Mobile node or Home agent functionality enabled in addition to Correspondent node. MN and HA can't be enabled at the same time. The last option turns on debugging messages for MIPL. Since MIPL is still work-in-progress you should enable this. With debug messages it is easier to figure out what is happening when something goes wrong.

After you finished the configuration, save changes and exit. Run `make dep` and compile and install the new kernel and modules. The kernel part is now done.

After the kernel part of MIPL is successfully installed and configured, you still have to compile and install the userspace tools. Run `configure` to create `Makefile` and `mobile-ip6` for your system. Run `make` and `make install` to compile and install userlevel tools, man pages, init scripts and example configuration files. These are mandatory for the module to work correctly. You also need to create the device file for MIPL with `mknod /dev/mipv6_dev c 0xf9 0`.

4.2. Configuration

The Mobile IPv6 configuration file can be found in `/etc/sysconfig/network-mip6.conf`. You can select from the following options for configuration:

- FUNCTIONALITY - Should this node act as a home agent (ha), mobile node (mn) or correspondent node (cn). HA and MN both have CN functionality embedded. Default value: cn.

- DEBUGLEVEL - In error situations it may be desired to get more detailed information what is happening. Increase this value to get more messages from the module (default: 0).
- TUNNEL_SITELOCAL - Should unicasts to node's site-local address be tunneled when mobile node is not in its home network (default: yes).
- MIN_TUNNEL_NR - Minimum number of free tunnel devices kept in cache on MN or HA. Must be set to at least 1 for MN and HA. To ensure successful bindings even during high work loads it could be set to a bigger value on the HA.
- MAX_TUNNEL_NR - Maximum number of free tunnel devices kept in cache on MN or HA. Must be set to at least 1 for MN and HA. To improve performance set it higher than MIN_TUNNEL_NR.
- HOMEDEV - Device where home address should be assigned to.
- HOMEADDRESS - Home address for mobile node with prefix length.

HOMEAGENT Home agent's address for mobile node with prefix length.

For run-time configuration and diagnostics we can use the `mipdiag` tool. An automatic startup script called `mobile-ip6` is included into the package. You can use `mobile-ip6 start` to start the module by hand and `mobile-ip6 stop` to unload. This script reads the configuration files and configures module accordingly. Another possibility is to load the module by hand using `insmod`. You cannot set Home Address nor Home Agent Address with `insmod` so Mobile Node will be left in a state where it does not know these addresses until given with the `mipdiag` tool.

If you want to use automatic module startup in RedHat, you must do `chkconfig mobile-ip6-level 345 on` that will setup all the necessary links. `mipdiag`, the diagnostic and configuration tool, is used to get statistics and state information and set runtime parameters.

5. Mobile IPv6 for Windows

Currently there is no full support publicly available for Mobile IPv6 in Windows operating systems. Microsoft offers CN support in Windows 2000/XP/2003. Additionally there is a different IPv6 stack for Windows, implemented by I2R (Institute for Infocomm Research), that offers CN support too (based on the recommendations from `draft-ietf-mobileip-ipv6-13.txt`).

We encountered difficulties during the installation of I2R for Windows 2000/XP so the trials are under progress. We tested the CN support from Windows 2003 (production release which integrates the Window 2000 IPv6 stack).

Whilst Windows 2000 IPv6 configuration is performed using `ipv6` tool, Windows 2003 has integrated it into the network configuration tool, called `netsh`. This is a shell that offers a set of several commands that resemble the router's configuration commands. The first step is to enter the network shell with the `netsh` command. Then you enter the `interface ipv6` mode to configure any IPv6 setting:

```
C:\> netsh
netsh> interface ipv6
```

The `set mobility` command permits us to configure the following mobile ipv6 parameters: `security`, `bindingcachelimit` and `correspondingnode`. The first specifies whether the BUs must be authenticated or not (the default values is enabled). We can also specify the number of binding cache entries with the `bindingcachelimit` parameter and the default value for the number of entries is 32. The last parameter, `correspondingnode`, specifies if the node will have CN capabilities. The default value for it is disabled. The enabling of CN permits the node to accept binding updates so route optimization can be performed.

```
netsh interface ipv6> set mobility
correspondingnode = enabled
netsh interface ipv6> set mobility
security = disabled
netsh interface ipv6> set mobility
bindingcachelimit = 1000
```

The `netsh` shell allows us to view the current setting and status with the following commands:

```
netsh interface ipv6>show mobility
netsh interface ipv6>show bindingcache
entries
```

The first command displays the values of the three parameters that we have configured and the second one displays the entries from the bindingcache, if there are any. We need to setup the default router for the CN to work properly.

```
netsh interface ipv6>set route <IPv6
address> /<integer> interface <IPv6
address>
```

where `<IPv6 address> /<integer>` is the route we want to set, `interface` is the name or index of the interface, `<IPv6 address>` is the gateway address. We can also display the routes that are already set up, with the following command:

```
netsh interface ipv6>show route
```

6. Trials

The trials performed focused on determining whether the Windows CN support will function with the Mobile IPv6 package from Linux and on a preliminary evaluation of the following parameters: RTT (Round Trip Time), inter-arrival jitter and cumulative number of packets lost. The trials were performed in our laboratory, which is part of CAMAN (Cluj-Napoca Academic Metropolitan Area Network).

6.1. Testbed Architecture

The trial architecture consisted of Fast Ethernet-based wired network with four workstations. The IPv6 router had three interfaces: one acted as Home Network (HN), whilst the others represented the Visited Networks (VN). On the VN interfaces we used `radvd` for stateless auto-configuration of the hosts connected to that network. You can see below the configuration for interface eth2:

```
interface eth2 {
    AdvSendAdvert on;
    prefix 3ffe:a:a:c::/64
           {
             AdvOnLink on;
             AdvAutonomous on;
             AdvRouterAddr on;
           };
};
```

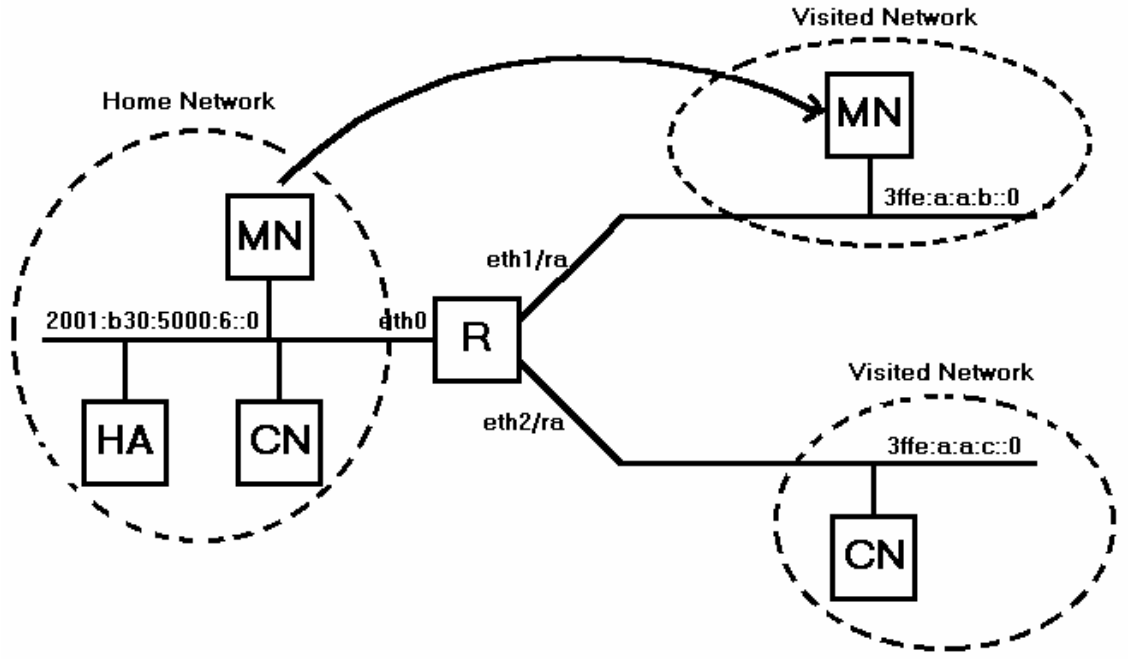

Figure 2. Trial testbed

The HA has an aggregatable global unicast address 2001:b30:5000:6::177/64. Below you will find the settings used in the mipl configuration file:

```
FUNCTIONALITY=ha
DEBUGLEVEL=4
TUNNEL_SITELOCAL=yes
MIN_TUNNEL_NR=1
MAX_TUNNEL_NR=3
```

The HA performs double tasks, acting also as correspondent node within the experiment. CN functionality is implemented together with HA or MN functionalities. The configuration of the MN does not differ from that of the HA. We must specify the home address of the MN, and the address of the HA. For the exact settings used you can see the lines below:

```
FUNCTIONALITY=mn
DEBUGLEVEL=1
HOMEADDRESS=2001:b30:5000:6::170/64
HOMEAGENT=2001:b30:5000:6::177/64
```

The movement of the MN from the home network to one of the visited networks, was accomplished by manually unplugging the network connector from one interface to another.

The fourth workstation, which operated under Windows 2003, was used for compatibility testing between the Linux and Windows operating systems, and for evaluating the effects of triangular routing.

For the first trials, the machine was configured to function in the home network of the MN. On the home link, it had an IPv6 address, a default gateway, set up and the CN operation was enabled.

The machine was then placed in the third network, and configured as in the first trials, but for addresses used.

Figure 2 presents the entire topology, with the machines placed in all the networks they were tested, although they operated only in one of them at a time.

6.2. Interoperability Testing

The testing of the Windows - Linux inter-operation took place while the MN was at first on the home link an MN that was later moved in one of the visited networks. The Windows-based CN machine sent `ICMPv6 echo request` messages to the home address of the MN. The results showed that for a brief period of time the connection between the two machines was lost (the period of time that the MN was not connected to any network), but shortly after that the connection was reestablished.

To find out if the CN has received the `binding update` message, from the MN, we used the netsh mobility command `show bindingcache`. We noticed that there was no entry in the cache, so we repeated the test, but this time we disabled the security option. There were again no entries in the binding cache.

In order to find out what the problem was, we used the Ethereal tool to capture all the traffic from the home link. After analyzing the packets, we discovered that the problem is caused by the different specifications of the two IETF recommendations. The Windows 2003 CN support is based on the 13th version, while the Linux Mobile IPv6 implementation is designed on the 19th version specifications. Furthermore the 19th recommendation specifies that, `Home Test Init`, `Home Test`, `Care-of Test Init` and `Care-of Test` messages are used to initiate the return routing procedure from the mobile node to the correspondent node. This ensures authorization of subsequent Binding Updates.

The results of the interoperability testing show that the two machines can communicate only with a packet tunneling performed by the HA, but this causes significant loss of performance.

6.3. Preliminary Evaluations

Our results focus mainly on RTT evaluation, while inter-arrival jitter and cumulative number of packets lost evaluations are in progress. The results in this paper were taken from the average RTT values, obtained by `ping6` utility. Note that it displays the maximum, average and minimum RTT. The influence of the packet size was studied too, varying from the default 56 bytes up to 50,000 bytes.

We conducted several trials and then we compared the results. The first thing we evaluated was the operation of the router from our laboratory compared to the one from the Communication Center. The following trial presents the differences between the MN operation on the home link and its operation in a visited network. To discover the delays cause by the Mobile IPv6 extensions we analyzed the times recorded when the MN operates in the visited network and when a computer is setup in one of the VN's. Our last measurements try to evaluate the effects caused by the lack of route optimization, the packets from the CN are tunneled to the MN by the HA.

Figure 3 presents the results provided by the testing of Router 1 (used in all of our trials) and Router 2 (i.e. IPv6 router from the Communication Center). The packet sizes varied from 56 up to 2048 bytes. As we can observe, the Router 1's performance is better than that of Router 2. The time it takes the router to forward a packet from one interface, is to be taken in consideration, in all of the following trials. Note that in the first scenario a Fast Ethernet hub connected all the hosts.

The following trial focuses on determining the differences between the RTT values for Mobile IPv6 and IPv6. This trial consisted of two parts. In the first part we determined the times between the HA and the MN, while the MN was in a visited network. The next step, was to configure in one of the visited networks a machine, that will act as an IPv6 node on that link. The RTT values were read once more. The two sets of values and their difference can be observed in figure 4. Although the time difference is fairly small, we must notice that RTT is bigger with 10% in average for Mobile IPv6 than for IPv6.

The time gap is caused by the IPv6 extensions used, by the time it takes the nodes to process the extra headers, like the routing header or the home address option.

The effects of packet tunneling are evaluated in the 4th series of trials. The RTT between the MN and the CN was determined in three situations. The first measurement was made when the CN was located in the 3rd test network. Then we configured the CN to function on the home link of the MN and we determined the RTT again. The final situation relied on the HA/CN embedded functionality offered by the MIPL implementation. While the last test used the Linux package the first two used the Windows 2003 support.

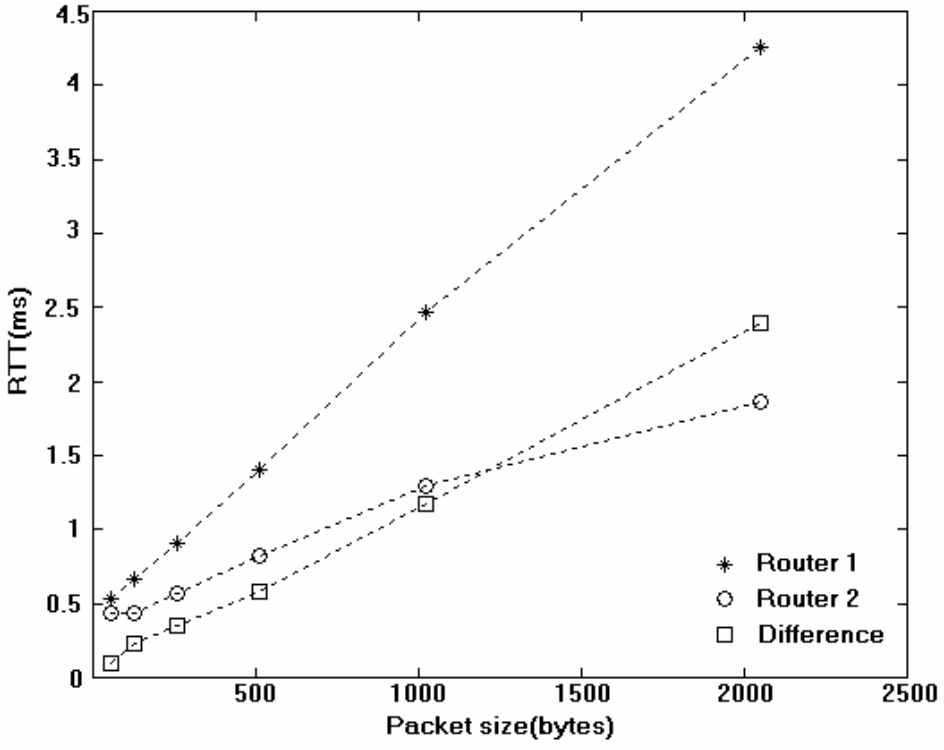

Figure 3. Router's performance

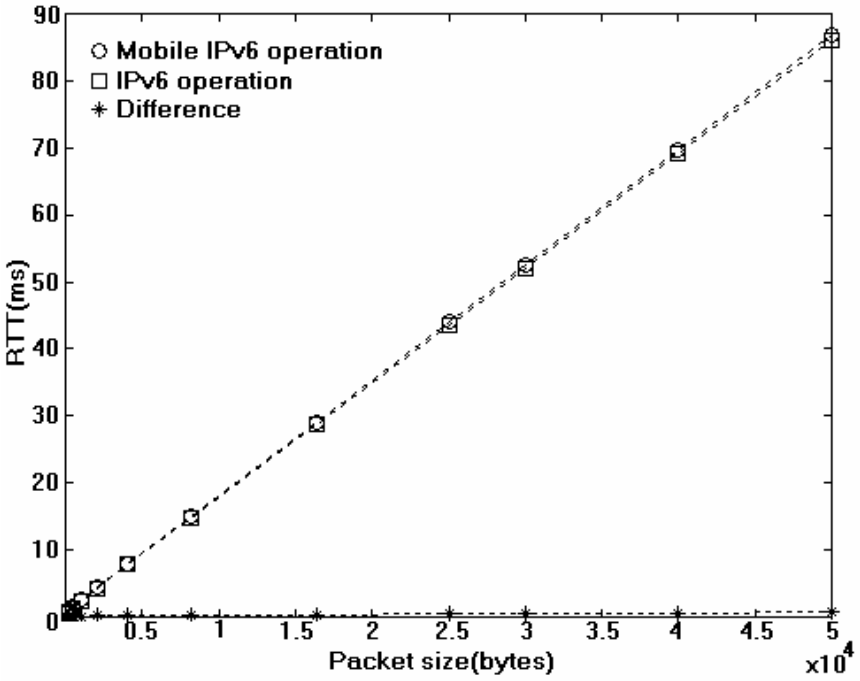

Figure 4. Mobile IPv6 – IPv6 comparison

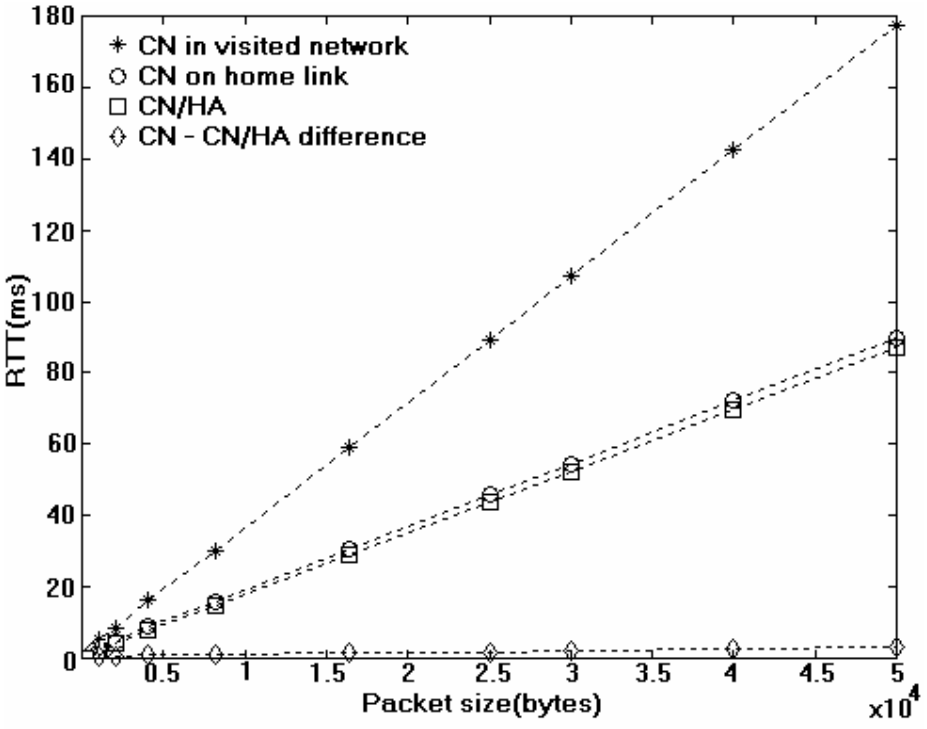

Figure 5. Packet tunneling effects

As we noticed when we tested the Linux-Windows interoperability, it is not possible for the CN to use the route optimization mechanism, so the packets addressed to the MN will be tunneled by the HA.

Figure 5 presents the results we have obtained, and also the differences between the packet tunneling operation mode and the routing header mechanism.

The results show that for small packets, up to 2048 bytes, the RTT for the packet tunneling is about 20% bigger than for routing header use. For bigger datagrams the value decreases, for 50000 bytes it reaches 4%.

The great RTT values recorded, when the CN was in the third network, demonstrate the necessity of route optimization. The times are over 100% greater than when routing header is used.

In all our trials the cumulative number packets loss was zero.

7. Conclusions and further work

The MIPL implementation is fully functional, as our trials have showed, but it is designed based on `draft-ietf-mobileip-ipv6-19.txt` although the current recommendation is `draft-ietf-mobileip-ipv6-21.txt`.

The Windows 2003 CN support does not function with the Linux package, because of the major differences between `draft-ietf-mobileip-ipv6-13.txt` and `draft-ietf-mobileip-ipv6-19.txt`. For a complete Windows-based Mobile IPv6 trial only, we have to wait until the proper recommendation will be released.

The preliminary RTT evaluation shows the importance of IPv6-enabled routers and the effects of route optimization. Several measurements related to the cumulative number of packet loss and inter-arrival jitter are under progress. Finally, the use of wireless devices is required for a complete evaluation of Mobile IPv6 operation.

8. References

[1] Johnson, D. and C. Perkins, *Mobility Support in IPv6. draft-ietf-mobileip-ipv6-21.txt*. 26 February 2003.

[2] Johnson, D. and C. Perkins, *Mobility Support in IPv6. draft-ietf-mobileip-ipv6-19.txt*. 29 Oct 2002.

[3] Johnson, D. and C. Perkins, *Mobility Support in IPv6. draft-ietf-mobileip-ipv6-13.txt*. 17 Nov. 2000.

[4] Johnson, D. and C. Perkins, *Route Optimization in Mobile IP. draft-ietf-mobileip-optim-11.txt.* 6 September 2001.

[5] Dobrota, V., *Digital Networks in Telecommunications. Vol. III: OSI and TCP/IP*, Second Edition, Mediamira Science Publishers, Cluj-Napoca, 2003 (in Romanian)

[6] Nikander, P. and C. Perkins, *Binding Authentication Key Establishment Protocol for Mobile IPv6. Draft-perkins-bake-01.txt.* 2 July 2001.